

Adabas Vista

Adabas Vista Programming Guidelines

Version 8.1.2

June 2008

This document applies to Adabas Vista Version 8.1.2 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © Software AG 2008. All rights reserved.

The name Software AG, webMethods and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. Other company and product names mentioned herein may be trademarks of their respective owners.

Table of Contents

1	Adabas Vista Programming Guidelines	1
2	Using Adabas Vista with Adabas	3
	Adabas Vista ISN	4
	Adabas Commands	5
	Adabas Parameters	6
	Adabas Security and Cipherring	7
	ETID	7
	Transaction Directives	7
	CID Handling	9
	ET Data	9
3	Additional Programming-Related Topics	11
	Coupled Files	12
	Command Limit	12
	Error Handling	12
	Multiclient File Processing	13
	Shared Partitions	13
	Single Partition Focus	14
	Updating the Partitioning Field	14
4	Adabas Vista Application Programming Interface	15
	Executing APIs with Natural and 3GL Programs	16
	API Function Overview	16
	API Function Descriptions	17
5	Using Adabas Vista with Adabas Cluster Services	19
	Clustered Applications	20
	Adabas Cluster Services	21
	Adabas System Coordinator	21
	Client Runtime Controls for Clustered Applications	21
6	Client Session Memory Requirements	23

1 Adabas Vista Programming Guidelines

This document provides information related to programming applications for use with Adabas Vista.

The following topics are provided:

- **Using Adabas Vista with Adabas**
- **Additional Programming-Related Topics**
- **Adabas Vista Application Programming Interface**
- **Using Adabas Vista with Cluster Services**
- **Client Session Memory Requirements**

2 Using Adabas Vista with Adabas

■ Adabas Vista ISN	4
■ Adabas Commands	5
■ Adabas Parameters	6
■ Adabas Security and Cipherring	7
■ ETID	7
■ Transaction Directives	7
■ CID Handling	9
■ ET Data	9

This section provides programming guidelines for using Adabas Vista with Adabas:

Adabas Vista ISN

If an Adabas file is defined as an Adabas Vista partitioned file, each Adabas ISN is modified with a partition identifier before returning the ISN to the application. Such a modified ISN is called an Adabas Vista ISN. Correspondingly, any ISN received from an application program for such a file is interpreted as an Adabas Vista ISN.

At the time the partition is defined, the space for a partition identifier is reserved within the 4-byte ISN field. The size of this area is based on the Adabas Vista file parameter `Maximum Number of Partitions`.

For example, the format of an Adabas Vista ISN for a partitioned file defined with a `Maximum Number of Partitions` set to 255 is as follows:

Offset	Length	Contents	Meaning
00	1	X'01' - X'FF'	The ID of the partition from which the record was retrieved.
01	3	X'000000' - X'FFFFFF'	The Adabas ISN of the record within the physical partition

Application programs that manipulate ISNs must be capable of handling the full 4-byte binary value returned. For Natural applications, any variable assigned to an ISN value should be defined with a minimum of P10. A value lower than this reduces the number of usable partitions and may result in a NAT1305 error code.



Caution: Special care is required when an application uses an ISN value as data. If a partitioned file is re-partitioned, the stored ISN may well become inaccurate.



Caution: When converting an Adabas file to a partitioned file, any ISNs for the file that may have been stored as data in other files must go through a conversion process to change the ISN values stored into correct Adabas Vista ISN values.

Adabas Commands

E1 Command (Refresh File)

If an `E1` command is issued with an ISN value of zero and with a blank Command ID, the file is refreshed. This is also true for a partitioned file.



Note: The `E1` command will be issued to each partition in turn. Any error which occurs when a later partition is refreshed (for example, Adabas response code 114 because the partition was not loaded with `PGMREFRESH=YES` specified) will result in a partially refreshed partitioned file.

S5 Command (Complex Find)

The `S5` command is only available for use with Adabas Vista partitioned files when the [Single Partition Focus](#) feature is enabled.

Record Hold Logic

When using Adabas Vista, it is recommended to use the `GET (L4)` command to hold records immediately before modifying them. This is particularly relevant to those programs that would otherwise use the `READ (L6)` and `FIND (S4)` commands against a partitioned file.

Refer also to the section Distributed Lock Mode.

Read Sequential Direction Changes

When issuing an `L9` command sequence, a second direction change is not allowed for distributed access.

When processing a direction change during an `L9` command sequence, Adabas Vista places restrictions on format and length overrides in the search buffer.

Release Global Format IDs

A `RC` command which uses the `C`, `L` or `O` command options in order to release specific or all global format IDs will only be issued to the database specified in the command. In addition, the command options will always be changed to `C` to ensure that all global format IDs generated by Adabas Vista are also released by Adabas.

Optional ISN Positioning

Any program that accesses a partitioned file and requires the use of ISN positioning (for example, a starting ISN on an initial`L3` command, or a minimum ISN value on an initial`Sx` command) must ensure that the supplied ISN is in Adabas Vista format. Adabas Vista then targets the appropriate partitions of the partitioned file.

Command ID within a Search Expression

Adabas Vista does not support a Command ID value enclosed in parentheses as a search expression for partitioned file access.

Adabas Parameters

The following Adabas ADARUN parameters may need to be tuned for use with Adabas Vista.

Adabas Parameter	Description
LI	<p>If a single file is partitioned into n files within the same database, it may obtain as many as n TBI elements per user as required for Adabas Vista distributed access.</p> <p>The increase in TBI elements depends on the amount of this distributed access and the number of partitions used in the average request. Therefore, the value of the <code>LI</code> parameter may need to be increased.</p>
LQ	The <code>LQ</code> parameter is similar to the <code>LI</code> parameter above. The <code>LQ</code> pool may need to be increased.
LU	The <code>LU</code> parameter value should be set to 65535 whether or not Adabas Vista is used. This requires a minimum <code>NAB</code> parameter setting of 16.
NAB	The <code>NAB</code> parameter value may need to be increased if the Adabas Vista read-ahead feature is used and/or a significant number of partitions are defined to the database.
NC	If one or more partitions are added to a database for the first time, the value for the <code>NC</code> parameter may need to be increased.
NH	See the <code>NISNHQ</code> parameter below.
NISNHQ	<p>Along with the <code>NH</code> parameter, this parameter may need to be increased if</p> <ul style="list-style-type: none">■ a significant number of new partitions are added to the database and hold-based commands are issued because of distributed access.■ the Adabas Vista read-ahead feature is used with hold-based commands.
NQCID	Similar to the <code>LI</code> parameter. The value for <code>NQCID</code> may need to be increased if a significant number of new partitions are added to the database and significant use of TBQ elements occurs because of distributed access.

Adabas Parameter	Description
NU	If one or more partitions are added to a database for the first time, the NU parameter value may need to be increased.
PREFETCH	This parameter is needed to control the Adabas Vista read-ahead feature at the job level. No other Adabas prefetch parameter is necessary.
TNAA, TNAE, TNAX, TT	Where partitions span more than one database, it is possible that a user may, during a series of related accesses, request numerous consecutive records from the same database within a distributed access. This may result in Adabas timing the user out on the other databases involved in the distributed access, which produces a response code 9 when Adabas Vista tries to continue the sequence on these databases. The values for the parameters TNAA , TNAE , TNAX, and TT should therefore be aggregated to the highest values for all databases containing partitions.

Adabas Security and Ciphering

When using Adabas Vista partitioned files, Adabas passwords and cipher codes must be the same for each partition.

ETID

Software AG strongly recommends that the use of ETID is consistent (identical) across all databases used for a client session.

Transaction Directives

Adabas Vista generates transaction directives in accordance with standard distributed transactions making it compliant with the use of Adabas Transaction Manager.

Correct distributed transaction processing is enforced by Adabas Vista (that is, an application is expected to be well-formed in the way it pursues transaction outcome). To explain...where multiple databases are modified the application is expected to pursue transaction outcome to all the changed databases, in series. For example, if two databases are modified then Adabas Vista expects to see two ET commands (one to each database) to cause a positive transaction outcome; or two BT commands (one to each database) for a negative transaction outcome. If an application tries to continue working after only issuing a single ET (or BT) sub-code 496 is the result. Similarly, if the application tries to modify another database without having fully issued all ET/BT commands for all databases in the previous transaction then sub-code 496 will also be issued. This behavior is compliant with distributed transaction processing standards, as implemented by Adabas Transaction Manager.

The runtime control `ET Data Database Number` is used to specify the location for transaction data.

Extended Hold

The P and M command options allow a program to commit (ET) or back out (BT) a transaction while leaving some records in hold status. Adabas Vista supports these options using the `Extended Hold` runtime control. Extended hold processing proceeds as follows:

- `Extended Hold = Minimum (default)`
 - When Adabas Vista receives the first transaction directive, it has no knowledge of any P or M options that are to be applied to the databases involved in the transaction except those specified on this first directive.
 - Adabas Vista therefore honors the P or M option specified for the source database number of the first directive. It then generates the necessary directives for all other source database numbers without regard to P or M options. Thus, the client's transaction is committed or rolled back as requested, P or M options on the first directive honored and all other ISNs released off hold.
- `Extended Hold = Maximum`
 - When Adabas Vista receives the first transaction directive, it has no knowledge of any P or M options that are to be applied to the databases involved in the transaction except those specified on this first directive.
 - Adabas Vista therefore honors the P or M option specified for the source database number of the first directive, but ensures that held records put on hold using other source databases are not released during the commit or roll-back process. Thus, the client's transaction is committed or rolled back as requested, but at this stage some of the transaction's held records may still be in hold status.

As each subsequent directive (in the sequence) is issued, Adabas Vista honours the P or M option specified for those source database numbers. When the Vista proxy detects "end of directive sequence", the proxy generates the necessary directives for each source database number that was not party to the client's directive sequence.

- If the application program fails to terminate the sequence of directives, records may unintentionally remain in hold status in one or more target databases. In this case, it is necessary to intervene manually by operator command or Adabas Online System to stop the client and free its resources; otherwise, the resources are freed when the client's session times out.

CID Handling

Translation and partitioning features mean that CID commands may be re-directed to other databases/files which can cause inadvertent clashing of CIDs inside Adabas databases. Consequently the CID handling in Adabas Vista has now been strengthened to avoid this happening.

ET Data

If an `ET` or `CL` command with ET data is issued by a client and, as a result of this, Vista has to generate a series of such commands, then Vista will always issue the ET data on the first `ET` or `CL` of the series. This is in accordance with standard distributed transaction protocols making it compliant with the use of Adabas Transaction Manager.

The runtime control Database Number for ET Data is used to specify the location for transaction data.

3

Additional Programming-Related Topics

■ Coupled Files	12
■ Command Limit	12
■ Error Handling	12
■ Multiclient File Processing	13
■ Shared Partitions	13
■ Single Partition Focus	14
■ Updating the Partitioning Field	14

This section provides additional information related to programming applications for use with Adabas Vista:

Coupled Files

Adabas Vista does not support search criteria that include syntax for physically coupled files or softly coupled files.

Command Limit

An optional command limit can be defined for a translation rule. If a client exceeds this limit then a response code 249, subcode 129 is returned. A command limit of 0 (zero) means that use of this particular rule is unlimited.

Error Handling

Adabas Vista uses the default Adabas response code 249 to indicate that an Adabas Vista processing error has occurred. This response code can be modified using the runtime control Error Response Code.

To qualify the error, a non-zero binary value is placed in the subcode field (the low-order 2-bytes of the Adabas Control Block Additions 2 field). Possible values for this subcode are provided in section Messages and Codes.

The application programmer can convert the response subcodes to error message text using the ERRM function of the Adabas Vista API (see [Application Programming Interface](#)).

When using Natural, the subcode may be retrieved using the USR0610N subprogram in library SYSEXT. This enables the following construct for an ON ERROR block:

```
0250  READ AVI1 LOGICAL BY NAME
0260      DISPLAY FIRST-NAME LAST-NAME SEX
0270  END-READ
0280  ON ERROR
0290  IF *ERROR-NR = 3249
0300      CALLNAT 'USR0610N' DB_ERR_STR
0310      WRITE 'AVI error, subcode:' DB_SUBCODE
0320  END-IF
0330  END-ERROR
0340  END
```

It is strongly recommended that error handling is implemented so that error conditions can be identified and handled immediately.

Refer to the runtime control `Error Reporting` for additional error reporting options.

Multiclient File Processing

Adabas Vista is able to process multiclient files. There are a few considerations that should be made when using this type of file with Adabas Vista:

- Adabas Vista does not support superuser Owner IDs with distributed access. The sequence of records returned will not be the same as for a normal Adabas multiclient file. These superusers are most likely to be used for utilities, which are not normally run with Adabas Vista.
- Adabas Vista cannot use the internal Owner ID as partition criteria. See the file parameter `Partitioning Field` for valid field types.

Shared Partitions

When processing Adabas Vista files that have some partitions shared, extra processing is required to determine the correct partition for a particular record, and to ensure that records are returned to the application in the most appropriate sequence for the command. This extra processing may have an adverse effect on performance for these command sequences.

For example, an `L1` command sequence processes all records in a file that is defined to Adabas Vista with three partitions, the first and third partitions sharing a single file. Adabas Vista will scan through the first file and extract all records that logically belong to the first partition. The second file will be processed as normal, and Adabas Vista will then scan through the first file again to extract the records that logically belong to the third partition.

When issuing an `L9` command against a shared partition where the descriptor used is not the partitioning field, although the count returned is correct, the ISN returned in the ISN Lower Limit field may indicate an incorrect partition. It is not possible to differentiate between the partitions in a shared file for an `L9` command.

Single Partition Focus

The Single Partition Focus feature can be used to define temporarily a partitioned file as having only one partition. Any command restrictions that may apply to partitioned files are lifted. Alternatively, you can set one of the partitions to FULL access and the others to NONE to achieve the same result.

You may enable single partition focus as the default for a partitioned file by using Adabas Vista Online Services. You may enable it for a single session by using the provided Adabas Vista API (example program API004UP).



Caution: If single partition focus is specified for a partition of a partitioned file, the validation of the partitioning field value against the defined criteria for that partition is no longer performed. This means that records whose partitioning field value does not correspond to the defined partitioning criteria may be updated or stored in that partition. If single partition focus is subsequently lifted and focused access performed against this partitioned file, such records may not be found. For this reason, it is recommended to restrict the use of single partition focus to those partitioned files for which no focused access is performed.

Example

A file is defined with two partitions and the partitioning field high value is set to M and Z, respectively. Single partition focus is specified for the first partition.

As a result, a record with the value T is added to the first partition.

When single partition focus is removed, a subsequent focused access for all records with the value T is directed to the second partition only.

Updating the Partitioning Field

Special consideration must be made if modifying the record's partition criteria will result in the movement of the record from one partition to another. First, it is inevitable that the ISN will change – which is absolutely not a normal programming situation – and which may require some application design consideration. Second, moving from one partition to another may require a distributed transaction which will only be safe if you use Adabas Transaction Manager. Adabas Vista will allow such record movement but only using a specific exit that defines how the record movement is to take place. Please contact Software AG in order to proceed in this area.

If no such exit is in use, any attempt to modify a record's partition criteria which would result in a cross partition move will return a response code. If such a modification is required, a DELETE and STORE operation must be performed.

4 **Adabas Vista Application Programming Interface**

■ Executing APIs with Natural and 3GL Programs	16
■ API Function Overview	16
■ API Function Descriptions	17

The following Adabas Vista interfaces are made available to the application programmer, some of which allow pre-defined configuration settings to be overridden. The programmer should be aware that the most efficient mode of operation for Adabas Vista is to use the pre-defined configuration settings and that any dynamic runtime changes by API may incur additional overhead.

All API functions are available for Natural and most API functions are also available for 3GL programs.

Executing APIs with Natural and 3GL Programs

The Adabas Vista installation INPL dataset includes example Natural programs for each of the available API functions. These Natural objects must be installed according to the instructions provided in the section Installation.

In a 3GL environment, the supplied AVIPAPI program enables a 3GL program to utilise these API functions. Each API function available for use by a 3GL program has an example program in the Adabas Vista source library. These programs are named APIASM nn , where nn corresponds to the Natural example program number in the table below. For example, APIASM09 is the sample 3GL API program for the CATEGORY function.

Any API invoked changes remain active for the duration of a client's session.

API Function Overview

The following table lists the available API functions:

Function	Option	Description	Example
CATEGORY	UPDATE	Modifies the client's current target category.	API009UP
	EXTRACT	Extracts the client's current target category.	
CLOG		Traces client commands (only if runtime control Trace is active).	API001UP / AVICLOG
CONVISN	ADATOAVI	Converts an ISN from Adabas format to Adabas Vista format (including Partition ID).	API007UP
	AVITOADA	Converts an ISN from Adabas Vista format to Adabas format.	
CRITREP		Lists partitions identified as unavailable for the specified partition file.	API005UP
ERRM		Translates a supplied Adabas Vista subcode into an error text message.	API002UP
PAGE	UPDATE	Modifies the client's current Page list.	API003UP

Function	Option	Description	Example
	EXTRACT	Extracts the client's current Page list.	
PARTLIST		Lists all partitions for the specified partitioned file.	API006UP
PARTOPTS	UPDATE	Modifies the client's current Access and Critical parameter values for the specified partitioned file and partition.	API004UP
	EXTRACT	Extracts the client's current Access and Critical parameter values for the specified partitioned file and partition.	
TRTARG	UPDATE	Modifies the target details of the client's translation rule identified by the supplied source database and file number.	API008UP
	EXTRACT	Extracts the target details of the client's translation rule identified by the supplied source database and file number.	

API Function Descriptions

- [CATEGORY Function](#)
- [PAGE Function](#)
- [PARTOPTS Function](#)
- [TRTARG Function](#)

CATEGORY Function

A client may identify a target category override to Adabas Vista using the CATEGORY function. This replaces the target category established for the client by the active runtime controls. Adabas Vista will use this new target category to direct application calls to the corresponding target database and file number.

The use of the UPDATE option of this function will cause a transparent tidy-up resulting in any active cursors, etc. being ended.

PAGE Function

A client may identify a page list override to Adabas Vista using the PAGE function. This replaces the page list established for the client by the active runtime controls. Adabas Vista will use this new page list to direct application calls to the corresponding target database and file number.

The use of the UPDATE option of this function will cause a transparent tidy-up resulting in any active cursors, etc. being ended.

PARTOPTS Function

The PARTOPTS function is used to update or extract the client's current `Access` and `Critical` partition parameter values for the specified partitioned file and partition.

The following `Access` values are available:

Value	Description
F	Set the <code>Access</code> partition parameter value to FULL.
R	Set the <code>Access</code> partition parameter value to READ.
N	Set the <code>Access</code> partition parameter value to NONE.
O	Enable the single partition focus feature for this partition.
X	Reset all <code>Access</code> partition parameter values to the defaults.

The following `Critical` values are available:

Value	Description
Y	Set the <code>Critical</code> partition parameter value to YES.
N	Set the <code>Critical</code> partition parameter value to NO.

TRTARG Function

This function can be used to extract or update the target details of a supplied source database and file number. This source database and file number is used in conjunction with the client's current target category and page list to identify the corresponding translation target details. An update request will fail if the source file is currently involved in a transaction. This function is only available for individual file translation rules and not "all-files" translation rules.



Caution: The use of the UPDATE option of this function means that the client session may be routing calls to a target database and file number which is different to that pre-defined in the configuration file.

5

Using Adabas Vista with Adabas Cluster Services

■ Clustered Applications	20
■ Adabas Cluster Services	21
■ Adabas System Coordinator	21
■ Client Runtime Controls for Clustered Applications	21

A clustered operating system is a collection of independent operating system images working together as one. The objective is to spread work around the cluster to achieve better load balancing, throughput, and availability.

Clustered applications appeared before their operating system counterpart. Application clusters operated over multiple processes within a single operating system image. Now clustered applications are enhanced to operate throughout an operating system cluster.

Adabas Cluster Services allows multiple instances of the Adabas server to operate against the same database across multiple operating system images, which enhances the general clustering approach for high-end systems.

Clustered Applications

A clustered application is a collection of independent jobs operating together as a single service.

A clustered application uses multiple jobs in unison to provide a single (clustered) service. CICS in MRO or plex mode, IMS/TM, and UTM are examples of such clustered applications.

Clustered applications can route transactions dynamically from one job (member) in the application cluster to another, usually for load-balancing purposes. Dynamic transaction routing can occur within an operating system image, or across operating system images that are members of an IBM sysplex cluster.

The Adabas System Coordinator provides Adabas Vista support for dynamic transaction routing within and across operating system images.

Each implementation of a clustered application allows client sessions to migrate from one job to another across one or more images in an operating system cluster. The memory associated with a particular client session must be migrated along with the client session itself.

The following clustered applications may be implemented within a single operating system image:

- CICS/MRO with dynamic transaction routing in z/OS and VSE/ESA
- IMS/TM in z/OS
- UTM in BS2000

The following clustered applications are available for operating system clusters:

- CICSplex in z/OS and
- IMSplex in z/OS

Adabas Cluster Services

Adabas Cluster Services allows you to use multiple images of the Adabas server across an operating system cluster in order to provide truly scalable throughput.

Adabas System Coordinator

Support for clustered applications within a single system image or across multiple system images in an operating system cluster is enabled by the Adabas System Coordinator when running one or more of the following Adabas add-on products:

- Adabas Fastpath
- Adabas Vista
- Adabas Transaction Manager
- Adabas SAF Security

Adabas Fastpath also uses the Adabas System Coordinator to host the Asynchronous Buffer Manager service.

Client Runtime Controls for Clustered Applications

When defining runtime controls for CICS/MRO with dynamic transaction routing or for CICSplex, select the job type CICS Cluster.

Clustered application job types have user context information allocated from shared memory rather than from local (Job) memory.

Additional fields are available to support clustered applications:




Note: For non-clustered applications, leave these fields empty.

Field	Used to...
System Coordinator Group Name	link the clustered application to the relevant Adabas System Coordinator group.
Clustered Application Service Name	group together all instances of the application to the Adabas System Coordinator.
Manage Terminal Sessions or Manage All Sessions	determine the type of sessions managed by the Adabas System Coordinator daemon.

Refer to the *Adabas System Coordinator* documentation for more information.

6 Client Session Memory Requirements

The following table lists the main memory requirements for an Adabas Vista client session.

 **Note:** This table is not intended to be an exhaustive list of the memory requirements. It is recommended that you use this table during the initial analysis of a system's memory requirements. Once implemented, Adabas System Coordinator Online Services may be used to more accurately monitor the amount of real memory in use.

Memory requirement (per session) for each...		Size (in bytes)
session area (one only)		512
database accessed		+ 160
file accessed	Adabas file	+ 80
	Adabas Vista translation file	+ 80 + 64 + 48
	Adabas Vista partitioned file	+ 80 + 64 + (48 + partitioning field length + partition concurrency * 5) * number of partitions
Command ID issued against	an Adabas file	+ 80
	an Adabas Vista translation file	+ 36 + 80
	an Adabas Vista partitioned file	+ (256 + 80 * number of involved partitions) + (length of Adabas buffers * number of involved partitions)
record held (within a transaction) against	an Adabas file	+ 64
	an Adabas Vista translation file	+ 64
	an Adabas Vista partitioned file	+ 64 * number of involved partitions

Example

1. Estimate the per session memory requirements for an application which has the following characteristics:

- two databases
- five Adabas files
- two Adabas Vista translation files
- one Adabas Vista partitioned file
 - Partitioning field length is eight bytes
 - User partition concurrency is eight
 - Number of partitions is three
 - Majority of access is distributed (all three partitions involved)
 - Average length of Adabas buffers is 1000 bytes
- Maximum Command ID concurrency of two (one Adabas file and one Adabas Vista partitioned file)
- Average number of successful hold commands between ET/BT is 20
 - 18 against Adabas files
 - two against Adabas Vista partitioned file

Result

User session area:		512
Database requirements:	$(2 * 160)$	320
File requirements:	$(5 * 80) + (2 * 192) + 1 * 432$	1216
Command ID requirements:	$80 + 3496$	3576
Held record requirements:	$18 * 64 + 2 * (64 * 3)$	1536
	Total	7160

2. Multiply this value by the estimated number of application clients to estimate the application memory requirements.
3. If a single TP system is host to multiple applications, repeat this process for each application to provide an overall estimate of the TP memory requirements.

The required memory is requested from either

- local TP memory using the host TP services; or
- shared memory using the Adabas System Coordinator daemon.

The use of shared memory depends on whether the job is defined as a clustered application using job types such as CICS Cluster, IMS, or UTM.



Note: It is strongly recommended that you discuss the use of such job types with your system programmer before you implement them.

